****

**Tecnologías de la Información**

[**Gael Fernando Gonzales C**](https://micampus.espe.edu.ec/user/view.php?id=4403&course=16496)**uasquer**

[**Jose Armando Meza R**](https://micampus.espe.edu.ec/user/view.php?id=11763&course=16496)**amirez**

**Carlos Sebastian Tonato Coronel**

**Tarea 07 Parcial 03**

**NRC: 22129**

**Fecha de entrega: 28-07-2025**

Contenido

[Manual de uso del Framework de HTML Materialize 3](#_Toc215171383)

[ **¿Qué es Materialize?** 3](#_Toc215171384)

[ Características Principales del Framework de diseño. 3](#_Toc215171385)

[ **Instalación e implementación.** 3](#_Toc215171386)

[ **Como Implementar Materialize “Local”, “Remoto”.** 5](#_Toc215171387)

[ **Materialize** 5](#_Toc215171388)

[ CSS de colores, en fondos y textos. 5](#_Toc215171389)

[ **Tipografía** 6](#_Toc215171390)

[ **Grid (Cuadriculas)** 7](#_Toc215171391)

[ **Helpers (ayudas)** 10](#_Toc215171392)

[ **Tablas** 14](#_Toc215171393)

[ **Media** 16](#_Toc215171394)

[ **Complementos de CSS Materialize** 20](#_Toc215171395)

[ **Botones** 21](#_Toc215171396)

[ **Footer** 21](#_Toc215171397)

[ **Iconos** 23](#_Toc215171398)

[ Nav-Bar 23](#_Toc215171399)

[ Bibliografía: 24](#_Toc215171400)

# Manual de uso del Framework de HTML Materialize

# **¿Qué es Materialize?**

Comúnmente conocido como Materialize CSS, es un framework de desarrollo web, específicamente orientado al front-end, incluye características de diseño de HTML, CSS y JavaScript.

# Características Principales del Framework de diseño.

Basado en Material Design: Utiliza conceptos de "papel y tinta", profundidad, sombras realistas, bordes redondeados, iconos y animaciones fluidas (como el efecto de "onda" al hacer clic en un botón).

Responsive (Adaptable): Viene con un sistema de rejilla (grid) de 12 columnas que hace que tu sitio web se adapte automáticamente a celulares, tabletas y computadoras de escritorio.

Componentes Prediseñados: Incluye elementos listos para copiar y pegar, como:

Barras de navegación (Navbars).

Adicionalmente Materialize incluye complementos, en esta versión de manual nos vamos a centrar en los CSS y omitiremos los JavaScript

* Tarjetas (Cards) para mostrar contenido.
* Botones flotantes de acción (FABs).
* Formularios estilizados.

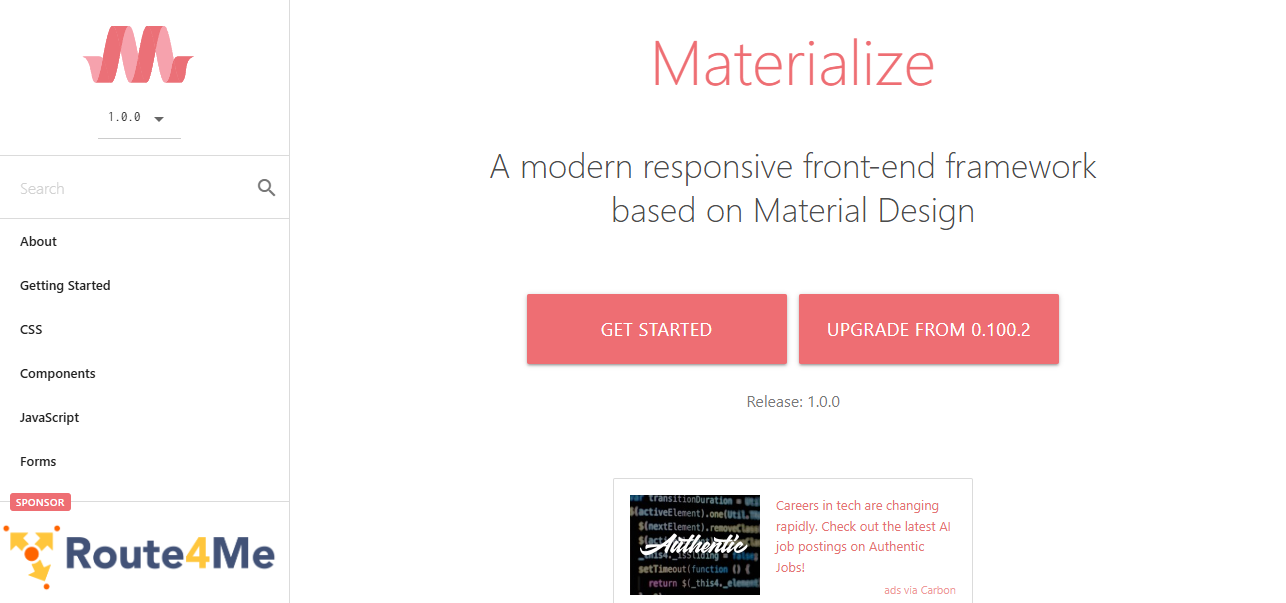
*Nota:*

JavaScript Integrado: Incluye funcionalidades interactivas como carruseles de imágenes, menús laterales (sidenavs), ventanas modales y efectos parallax.

# **Instalación e implementación.**

Antes de abordar el framework de diseño como tal, mostraremos una pequeña guía de instalación e implementación del framework en HTML.

Nos dirigimos a la página web oficial: <https://materializecss.com>



En la sección de descargas encontraremos diferentes enlaces que nos ayudaran a descargar diferentes complementos del proyecto Materialize a continuación explicamos para que sirve cada uno y el recomendado va a ser el CDN.



En el apartado de descargas encontramos 3 opciones:

* Materializar (Esta opción nos sirve para descargar todo el proyecto de framework que incluye los archivos de CSS y JavaScript tanto editables como no editables y trabajar de manera local).
* Sass (Esta opción nos sirve para los archivos Sass).
* CDN (Esta opción nos sirve para trabajar con el framework de manera online).

# **Como Implementar Materialize “Local”, “Remoto”.**

Forma Local Materializar

Para implementar de forma local se debe hacer los enlaces a las carpetas donde se descargó el proyecto, por ejemplo.

|  |
| --- |
| <!-- Compiled and minified CSS -->  <link rel="stylesheet" href="Plantilla/css/materialize/[materialize.min.css](http://materialize.min.js)">  <!--JavaScript at end of body for optimized loading→  <link rel="stylesheet" href="Plantilla/css/materialize/[materialize.min.js](http://materialize.min.js)"> |

Forma Online CDN

Se debe implementar estas líneas de enlace que nos proporciona la pagina oficial de materialize.

|  |
| --- |
| <!-- Compiled and minified CSS -->  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">  <!-- Compiled and minified JavaScript →  <script src="<https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js>">  </script> |

# **Materialize**

# CSS de colores, en fondos y textos.

Paleta de colores, fondos y textos coloridos:

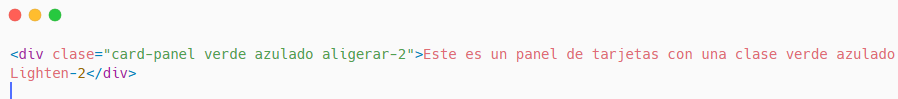
Materialize incluye una paleta de colores basada en los colores base del diseño del material del framework. Cada uno de estos colores se define con una clase de color base y una clase opcional de aclarar u oscurecer.

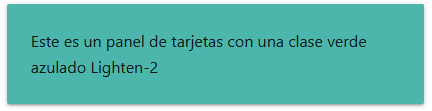
* Color Base: red, blue, green, orange, teal, etc.
* Variante:

1. lighten (más claro, mezclado con blanco).
2. darken (más oscuro, mezclado con negro).
3. accent (colores brillantes y saturados para destacar botones o alertas).
4. Intensidad: Un número del 1 al 4 (o hasta el 5 en el caso de lighten).

Dentro de Materialize existen varias paletas de colores y posibles combinaciones de estas la cuales se puede consultar en la página oficial de Materialize en el siguiente enlace: <https://materializecss.com/color.html> .

Colores de fondo: Para aplicar un color de fondo, simplemente agregue el nombre del color y la luz/oscuridad como clase al elemento.





Colores de texto: Para aplicar un color de texto, simplemente agregue -text a la clase de color así:

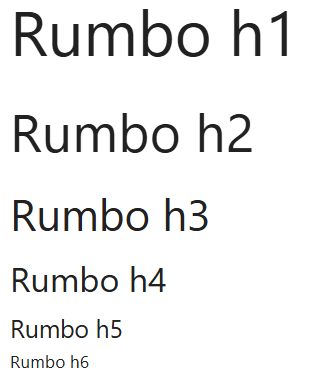
Imagen que contiene Diagrama

El contenido generado por IA puede ser incorrecto.



# **Tipografía**

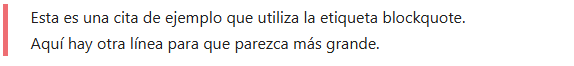
Encabezados: Proporcionamos un estilo básico en las etiquetas de encabezado. En el ejemplo, puede ver los diferentes tamaños de las 6 etiquetas de encabezado.



Frases de bloque: Las comillas en bloque se utilizan principalmente para dar énfasis a una cita o cita. También puedes usarlos para obtener más jerarquía y énfasis en el texto.

Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.



Texto de flujo: Un defecto común que hemos visto en muchos marcos es la falta de soporte para texto verdaderamente responsivo. Si bien los elementos de la página cambian de tamaño de manera fluida, el texto aún cambia de tamaño de forma fija. Para mejorar este problema, para páginas con mucho texto, hemos creado una clase que escala de manera fluida el tamaño del texto y la altura de la línea para optimizar la legibilidad para el usuario. La longitud de la línea se mantiene entre 45 y 80 caracteres y la altura de la línea aumenta en pantallas más pequeñas.

Nombre de la empresa

El contenido generado por IA puede ser incorrecto.

# **Grid (Cuadriculas)**

Estamos utilizando un sistema de rejilla sensible a fluidos estándar de 12 columnas. La cuadrícula le ayuda a diseñar su página de forma ordenada y sencilla. Nuestra cuadrícula estándar tiene 12 columnas. No importa el tamaño del navegador, cada una de estas columnas siempre tendrá el mismo ancho.

Para tener una idea de cómo se utiliza la cuadrícula en HTML, eche un vistazo al código a continuación, que producirá un resultado similar al anterior.



Compensaciones: Las compensaciones según Materialize son prefijos para dar el tamaño a las columnas en cada fila las cuales se explican a continuación.

Para compensar, simplemente agregue compensación-s2 a la clase donde s significa el prefijo de clase de la pantalla (s = pequeño, m = medio, l = grande) y el número siguiente es el número de columnas que desea compensar.

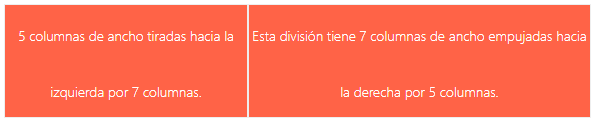


Empujar y estirar: Esto nos sirve para dar un tamaño más dinámico a las columnas dentro de los contenedores

Puedes cambiar fácilmente el orden de tus columnas empujando y tirando. Simplemente agregue push-s2 o pull-s2 a la clase donde s significa el prefijo de clase de la pantalla (s = pequeño, m = medio, l = grande) y el número que sigue es el número de columnas que desea empujar o tirar.

Texto

El contenido generado por IA puede ser incorrecto.



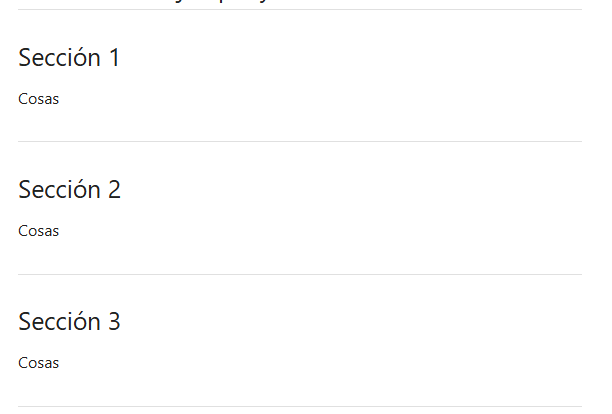
Divisor: Los divisores son líneas de 1 píxel que ayudan a dividir su contenido.

Sección: La clase sección se utiliza para acolchado superior e inferior

Ejemplo:

Texto

El contenido generado por IA puede ser incorrecto.



# **Helpers (ayudas)**

Estas clases son herramientas rápidas para ajustar alineaciones, ocultar elementos o dar efectos sin escribir una sola línea de CSS personalizada.

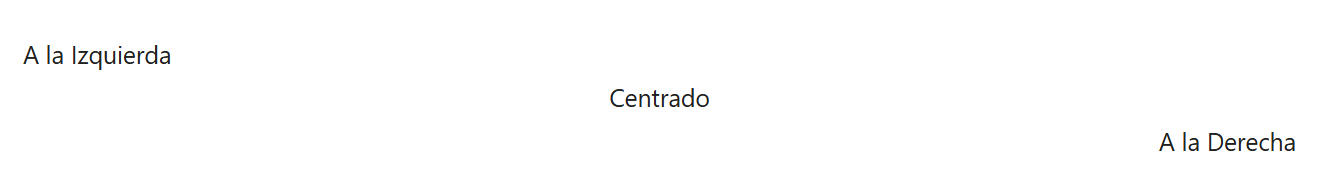
1. Alineación de Texto (Alignment)

Estas clases sirven para alinear texto dentro de su contenedor. Equivalen a usar text-align en CSS.

* .left-align: Alinea el texto a la izquierda.
* .right-align: Alinea el texto a la derecha.
* .center-align: Centra el texto.
* .justify-align: Justifica el texto (útil para párrafos largos).

Ejemplo:





2. Alineación Vertical (valign-wrapper)

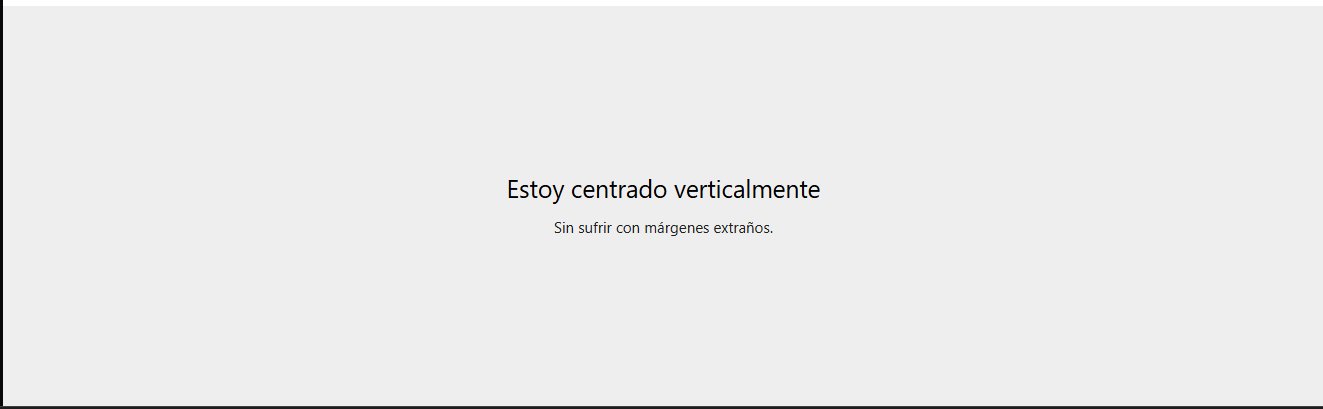
Esta es una de las clases más buscadas. Sirve para centrar contenido verticalmente dentro de un contenedor.

Requisito: El contenedor debe tener la clase .valign-wrapper.

Ejemplo:

Texto

El contenido generado por IA puede ser incorrecto.



3. Ocultar Contenido (Hiding Content)

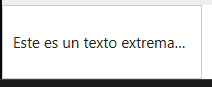
Materialize ofrece clases para ocultar elementos según el tamaño de la pantalla (Responsive Design). Esto es vital para que tu sitio se vea bien en celulares y PCs.

|  |  |
| --- | --- |
| Clase | Comportamiento |
| .hide | Oculto en todos los dispositivos. |
| .hide-on-small-only | Oculto solo en Móviles (aprox < 600px). |
| .hide-on-med-only | Oculto solo en Tablets (aprox < 992px). |
| .hide-on-med-and-down | Oculto en Tablets y Móviles. |
| .hide-on-med-and-up | Oculto en Tablets y PC. |
| .hide-on-large-only | Oculto solo en PC. |

4. Truncar Texto (truncate)

Si tienes una línea de texto que es demasiado larga y no quieres que rompa el diseño (por ejemplo, en una tarjeta de presentación), usas esta clase. Corta el texto y agrega puntos suspensivos (...).

Ejemplo

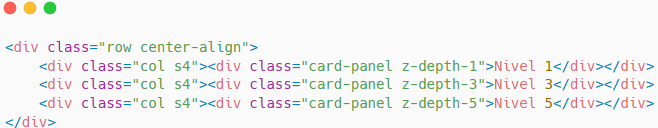


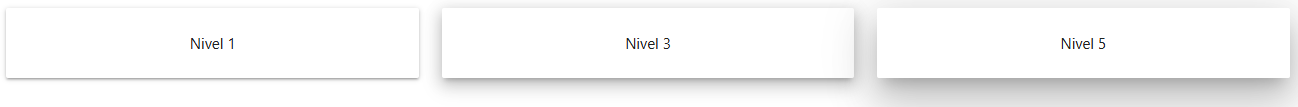
5. Sombras y Profundidad (z-depth)

Material Design se basa en la profundidad (como si fueran hojas de papel una encima de otra). Puedes controlar qué tan "elevado" parece un elemento con las sombras.

Va del 0 (sin sombra) al 5 (sombra muy pronunciada).

Ejemplo:





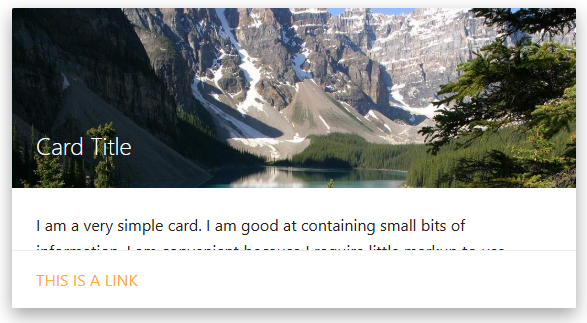
6. Efecto Hover (hoverable)

Esta clase añade una transición de sombra cuando pasas el mouse por encima del elemento. Es muy común usarlo en las tarjetas (cards) para indicar que son "clicables".

Ejemplo:

Texto

El contenido generado por IA puede ser incorrecto.



7. Floats (Flotantes)

Para empujar elementos rápidamente a los extremos.

* .left: Flota a la izquierda (float: left !important).
* .right: Flota a la derecha (float: right !important).

Ejemplo en un botón:

Logotipo, nombre de la empresa

El contenido generado por IA puede ser incorrecto.



8. Pulse

Llama la atención sobre tus botones con este efecto sutil pero cautivador. Simplemente agregue la clase pulse a tu botón. Nota: Esto está destinado a botones flotantes, por lo que es posible que no funcione perfectamente con todos los componentes.

Texto

El contenido generado por IA puede ser incorrecto.



# **Tablas**

|  |  |
| --- | --- |
| Clase | Función |
| .striped | Rayada: Añade un color de fondo gris suave a las filas impares (efecto cebra). Ayuda mucho a leer tablas largas. |
| .highlight | Resaltado: Cuando pasas el mouse por encima de una fila, esta cambia de color ligeramente. Ideal para indicar qué fila estás mirando. |
| .centered | Centrado: Alinea todo el texto (encabezados y celdas) al centro. Por defecto, Materialize alinea a la izquierda. |
| .responsive-table | Adaptable: Fundamental para móviles. Si la tabla es muy ancha, permite hacer scroll horizontal dentro de la tabla sin romper el diseño de la página. |





# **Media**

El objetivo principal de estas clases es hacer que los elementos multimedia sean responsivos (que se adapten al tamaño de la pantalla) sin que tengas que pelear con anchos fijos o desbordamientos.

1. Imágenes (Images)

Materialize ofrece dos clases principales para el manejo de imágenes con puro CSS.

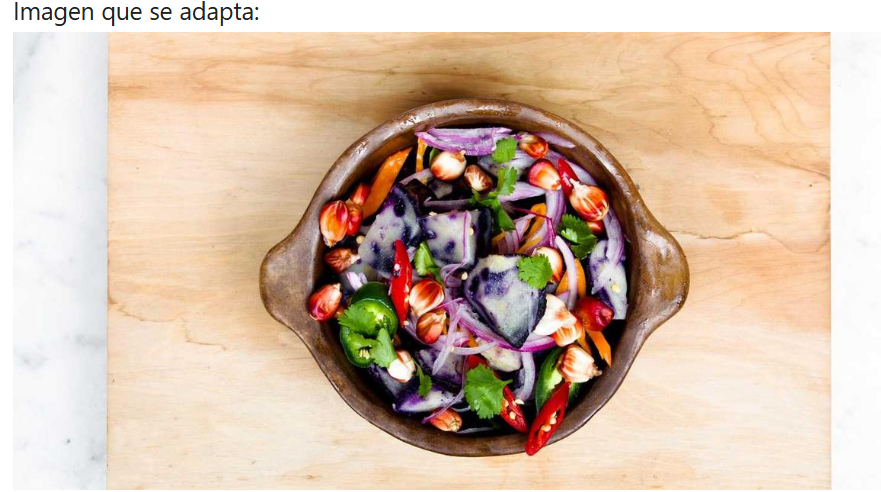
A. Imágenes Responsivas (.responsive-img)

Por defecto, si pones una imagen gigante en un celular, se saldrá de la pantalla. Esta clase ajusta automáticamente la imagen al ancho de su contenedor.

Lo que hace: Establece max-width: 100% y height: auto.

Ejemplo:





B. Imágenes Circulares (.circle)

Muy usada para fotos de perfil, avatares o testimonios.

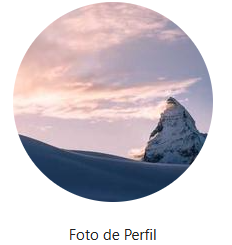
Lo que hace: Aplica un border-radius: 50%.

Nota: Para que se vea un círculo perfecto, la imagen original debería ser cuadrada; si es rectangular, se verá como un óvalo.

Ejemplo:

Logotipo, nombre de la empresa

El contenido generado por IA puede ser incorrecto.



2. Videos (Videos)

Manejar videos en diseño web (especialmente de YouTube o Vimeo) suele ser difícil porque tienen alturas fijas que rompen el diseño en móviles. Materialize soluciona esto con contenedores.

A. Videos Embebidos (.video-container)

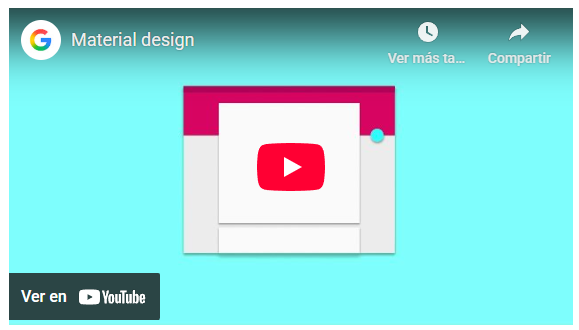
Esta es la clase mágica para videos de YouTube o Vimeo (elementos <iframe>).

Cómo se usa: Debes envolver el iframe dentro de un div con la clase .video-container.

El truco: Crea una caja con relación de aspecto 16:9 que se escala perfectamente.

Ejemplo:





B. Videos HTML5 (.responsive-video)

Si no usas YouTube, sino que tienes un archivo de video propio (.mp4) cargado con la etiqueta <video>, usas esta clase. Funciona igual que la de las imágenes.

Ejemplo:



Texto

El contenido generado por IA puede ser incorrecto.

Nota sobre JavaScript

En la documentación oficial de Media existe un componente llamado "Material Box" (que hace un efecto de zoom tipo "lightbox" al hacer clic en una imagen).

# **Complementos de CSS Materialize**

Las Cards (Tarjetas) son posiblemente el componente más utilizado en Materialize porque sirven para casi todo: mostrar artículos de blog, perfiles de usuario, productos o noticias.

Materialize ofrece varias configuraciones. Aquí tienes las 3 variantes más importantes que funcionan perfectamente solo con CSS.

1. La Estructura Anatómica

Una tarjeta se construye como un bloque de Lego. No es obligatorio usar todas las piezas, pero este es el orden lógico:

* .card: El contenedor principal.
* .card-image (Opcional): Para fotos destacadas.
* .card-content: Donde va el texto principal (p). Tiene padding (espacio interno) automático.
* .card-action (Opcional): Una franja en la parte inferior para enlaces o botones. Tiene un borde superior sutil para separarlo del contenido.

2. Tarjeta con Imagen (La Clásica)

Esta es la tarjeta estándar. Nota cómo la clase. card-title se coloca dentro de la imagen para que el texto aparezca sobre la foto con una sombra sutil para leerse mejor.





# **Botones**

Existen tres tipos principales de botones descritos en Material Design. El botón en relieve es un botón estándar que indica acciones y busca dar profundidad a una página mayormente plana. El botón de acción circular flotante está diseñado para funciones muy importantes. Los botones planos se suelen usar en elementos que ya tienen profundidad, como tarjetas o modales.

Cada uno de los botones es personalizable

Aumentó:





Flotante:

Icono

El contenido generado por IA puede ser incorrecto.



Departamento:





# **Footer**

El Footer (Pie de página) en Materialize es muy fácil de implementar. Está diseñado para ser robusto y visualmente separado del resto del contenido.

* La clase mágica aquí es .page-footer.

Aquí tienes la guía para crear un footer estándar y, lo más importante, cómo hacer que se quede siempre abajo (Sticky Footer), que es el problema más común al diseñar webs.

El **Footer** (Pie de página) en Materialize es muy fácil de implementar. Está diseñado para ser robusto y visualmente separado del resto del contenido.

La clase mágica aquí es .page-footer.

Aquí tienes la guía para crear un footer estándar y, lo más importante, cómo hacer que se quede siempre abajo (Sticky Footer), que es el problema más común al diseñar webs.

1. La Estructura Básica

Un footer en Materialize se divide en dos secciones principales dentro de la etiqueta <footer>:

Contenido Principal: Donde pones los enlaces, la biografía o información de contacto. Usa el sistema de rejillas (row y col).

Footer Copyright: Una barra inferior con un color ligeramente diferente (generalmente más oscuro) para el aviso de derechos de autor.

Las clases clave:

Texto

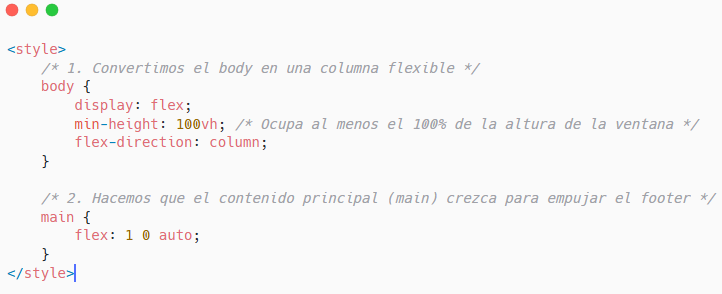
El contenido generado por IA puede ser incorrecto.

2. El Truco del "Sticky Footer" (Footer Pegajoso)

Un problema muy común en desarrollo web es que si tu página tiene poco contenido (ej: solo un párrafo), el footer sube y se queda flotando a mitad de la pantalla, dejando un espacio blanco feo abajo.

Para evitar esto y obligar al footer a estar siempre al final visualmente, Materialize recomienda usar Flexbox en el body.

Debes añadir este pequeño fragmento de CSS a tu archivo o en el <head>:



# **Iconos**

Materialize incluye 932 íconos de Material Design cortesía de Google. Puedes descargarlos directamente desde las especificaciones de Material Design.



Algunos iconos incluidos

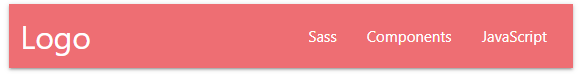
Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

# Nav-Bar

El Navbar (Barra de Navegación) es la carta de presentación de tu sitio web. En Materialize, crear una barra moderna, con sombras y colores vibrantes, es cuestión de pocas líneas de código.

La etiqueta principal siempre es <nav> y dentro de ella SIEMPRE debes poner un div con la clase .nav-wrapper.



Texto

El contenido generado por IA puede ser incorrecto.

**1. Posición del Logo**

La clase .brand-logo es la encargada de posicionar tu título o imagen. Por defecto, se alinea a la izquierda, pero puedes cambiarlo fácilmente:

* **Izquierda (Por defecto):** <a href="#" class="brand-logo">Logo</a>
* **Derecha:** <a href="#" class="brand-logo right">Logo</a>
* **Centro:** <a href="#" class="brand-logo center">Logo</a>

**Importante:** Si usas brand-logo center, asegúrate de que tus enlaces (ul) no se solapen con el logo en pantallas medianas.

Texto

El contenido generado por IA puede ser incorrecto.

# Bibliografía:

Materialize. (s.f.). *Materialize: A modern responsive front-end framework based on Material Design*. Recuperado el 27 de noviembre de 2025, de <https://materializecss.com>